

21장 빌트인 객체

객체의 분류

- **표준 빌트인 객체(Standard built-in objects)**

표준 빌트인 객체는 ECMAScript 사양에 정의된 객체를 말하며, 애플리케이션 전역의 공통 기능을 제공한다.

- **호스트 객체(host objects)**

호스트 객체는 ECMAScript 사양에 정의되어 있지 않지만 자바스크립트 실행 환경(브라우저 환경 또는 Node.js 환경)에서 추가로 제공하는 객체를 말한다.

- **사용자 정의 객체(user-defined objects)**

사용자 정의 객체는 표준 빌트인 객체와 호스트 객체처럼 기본 제공되는 객체가 아닌 **사용자가 직접 정의한 객체**를 말한다.

표준 빌트인 객체

math, Reflect, JSON을 제외한 표준 빌트인 객체는 모두 인스턴스를 생성할 수 있는 생성자 함수 객체다.

생성자 함수 객체인 표준 빌트인 객체는 프로토타입 메서드(Object.prototype.toString)와 정적 메서드(Object.create 등)를 제공하고, 생성자 함수 객체가 아닌 표준 빌트인 객체는 정적 메서드만 제공한다.

원시값과 래퍼 객체

```
const str = 'hello';

// 원시 타입인 문자열이 프로퍼티와 메서드를 갖고 있는 객체처럼 동작한다.
console.log(str.length); // 5
console.log(str.toUpperCase()); // HELLO
```

위 예제와 같이 `str`은 원시값인 문자열을 담은 변수인데, `str.length`, `str.toUpperCase()`와 같이 프로퍼티와 메서드를 호출할 수 있다.

이는 원시값에 객체처럼 **마침표 표기법**으로 접근하면 자바스크립트 엔진이 **일시적으로 원시값을 연관된 객체로 변환**해 주기 때문이다.

즉, 원시값을 객체처럼 사용하면 자바스크립트 엔진은 암묵적으로 연관된 객체를 생성하여 생성된 객체로 프로퍼티에 접근하거나 메서드를 호출하고 다시 원시값으로 되돌린다.

이러한 임시 객체를 래퍼 객체라 칭한다

래퍼객체는 처리가 종료된 후 가바지 컬렉션의 대상이 되어 사라진다

전역 객체

전역 객체(`global object`)는 코드가 실행되기 이전 단계에 자바스크립트 엔진에 의해 어떤 객체보다도 먼저 생성되는 특수한 객체이며, 어떤 객체에도 속하지 않은 최상위 객체다.

특징

- 전역 객체는 개발자가 의도적으로 생성할 수 없다. 즉, 전역 객체를 생성할 수 있는 생성자 함수가 제공되지 않는다.
- 전역 객체의 프로퍼티를 참조할 때 `window` (또는 `global`)를 생략할 수 있다.
- 전역 객체는 `Object`, `String`, `Number`, `Boolean` 등 모든 표준 빌트인 객체를 프로퍼티로 가지고 있다.
- 자바스크립트 실행환경(브라우저 또는 Node.js)에 따라 추가적인 프로퍼티와 메서드를 갖는다.(브라우저: DOM, BOM, Canvas, XMLHttpRequest등 / Node.js: Node.js 고유의 API)
- `var` 키워드로 선언한 전역 변수, 선언하지 않은 변수에 값을 할당한 암묵적 전역, 전역 함수는 전역 객체의 프로퍼티가 된다.

빌트인 전역 프로퍼티

- **Infinity**
 - 무한대를 나타내는 숫자값
- **NaN**

- 숫자가 아님(`Not a Number`)을 나타내는 숫자값 `NaN`
- **undefined**
 - 원시 타입 `undefined`

빌트인 전역 함수

- **eval**
 - 함수는 자바스크립트 코드를 나타내는 문자열을 인수로 전달받아, 전달받은 문자열 코드가 표현식 이라면 함수는 문자열 코드를 런타임에 평가하여 값을 생성하고,
 - 전달받은 인수가 표현식이 아닌 문이라면 함수는 문자열 코드를 런타임에 실행한다.
 - `eval` 함수는 보안에 매우 취약하다.
또한, 최적화가 수행되지 않아 일반적인 코드 실행에 비해 느리다.
따라서 `eval` 함수의 사용은 금지해야 한다.
- **isFinite**
 - 전달받은 인수가 정상적인 유한수 인지 검사하여 유한수이면 `true`, 무한수이면 `false` 를 반환
- **isNaN**
 - 전달받은 인수가 `NaN` 인지 검사하여 그 결과를 불리언 타입으로 반환한다.
 - 전달받은 인수의 타입이 숫자가 아닌경우 숫자로 타입을 변환한 후 검사를 수행한다.
- **parseFloat**
 - 전달받은 문자열 인수를 실수로 해석하여 반환한다.
- **parseInt**
 - 전달받은 문자열 인수를 정수로 해석하여 반환한다.
- **encodeURIComponent / decodeURI**
 - 완전한 `URI Uniform Resource Identifier` 를 문자열로 전달받아 이스케이프 처리를 위해 인코딩한다.
 - 이스케이프 처리

- 네트워크를 통해 정보를 공유할 때 어떤 시스템에서도 읽을 수 있는 **아스키 문자 셋**으로 변환하는 것이다. 예를들면 한글 '가'는 %EC%9E%90 으로 인코딩된다.
- **encodeURIComponent / decodeURIComponent**
 - **URI** 구성요소를 인수로 전달받아 인코딩한다

암묵적 전역

선언하지 않은 식별자에 값을 할당하면 전역 객체의 프로퍼티로 등록되는걸 의미한다.

Refernece

- <https://hong-p.github.io/javascript/javascript-deepdive-ch21/>